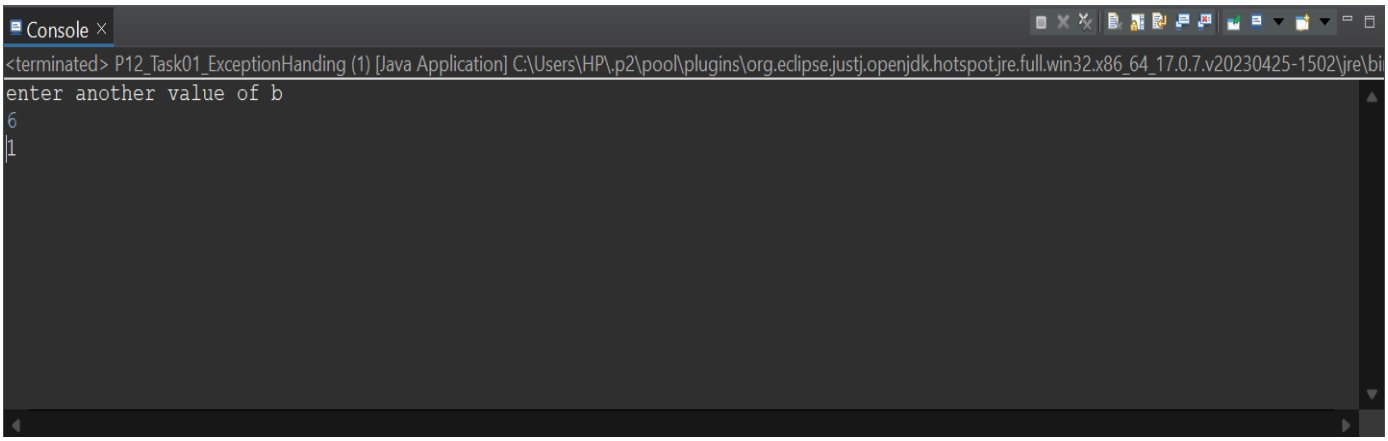
Practical-12

```
package unit03;
import java.util.*;
public class P12_Task01_ExceptionHanding {

	public static void main(String[] args) {
		FunctionsOfException obj = new FunctionsOfException();
		obj.handleException();
		//obj.inputMismatchException();
		//obj.stackOverFlowError();
		//obj.indexOutOfBoundException();
		//obj.nullPointerException();
	}

}
class FunctionsOfException{
	Scanner scan = new Scanner(System.in);
	void basicException() {
		int a=9;
		int b=0;
		//System.out.print(a/b);
	}
	void handleException() {
		int a=9;
		int b=0;
		try {
			int c=a/b;
			System.out.println(c);
		}catch(Exception e) {
			System.out.println("enter another value of b");
			b=scan.nextInt();
			int c=a/b;
			System.out.println(c);
		}
	}
	void multipleCatch() {
		int a=9;
		int b=0;
		try {
			int c=a/b;
			System.out.println(c);
		}catch(ArithmeticException e) {
```

```
                System.out.println("catch by arithmeticException enter another value
of b");
                b=scan.nextInt();
                int c=a/b;
                System.out.println(c);
            }
            catch(RuntimeException e) {
                System.out.println("catch by RuntimeException enter another value of
b");
                b=scan.nextInt();
                int c=a/b;
                System.out.println(c);
            }
            catch(Exception e) {
                System.out.println("enter another value of b");
                b=scan.nextInt();
                int c=a/b;
                System.out.println(c);
            }
        }
        void inputMismatchException() {
            try{
                int a=scan.nextInt();
                System.out.println(a);
            }catch(InputMismatchException e) {
                System.out.println("enter integer value!!");
                scan.nextLine();
                int a=scan.nextInt();
                System.out.println(a);
            }
        }
        void stackOverFlowError() {
            int i=1;
            while(i>0) {
                stackOverFlowError();
                i++;
            }
        }
        void indexOutOfBoundException() {
            try{
                int a[]= {1,2,3};
            System.out.println(a[3]);
            }catch(Exception e) {
                System.out.println("indexOut Of Bound Exception");
```

```
                int a[]= {1,2,3};
                System.out.println(a[2]);
            }
        }
        void nullPointerException() {
            try{
                String a=null;
                System.out.println(a.length());
            }catch(Exception e) {
                System.out.println("null Pointer Exception");
                String a=scan.nextLine();
                System.out.println(a.length());
            }
        }

}
```

Output

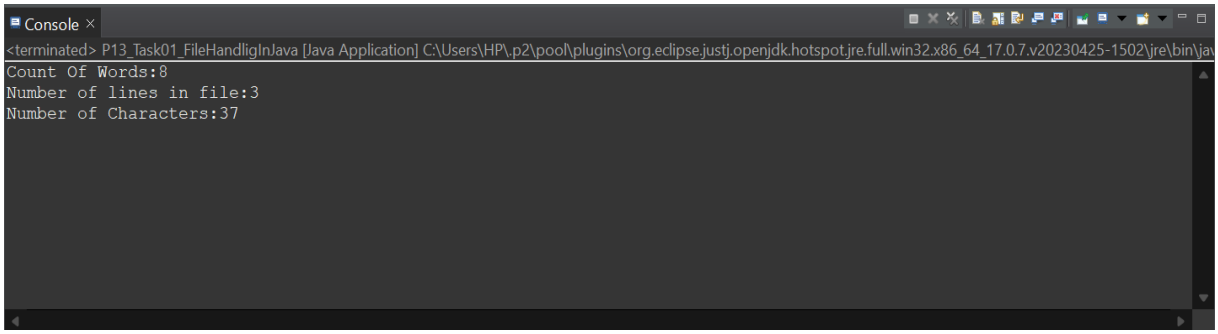

Console
<terminated> P12_Task01_ExceptionHanding (1) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bi
enter another value of b
6
1

Practical-13

```
package unit03;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class P13_Task01_FileHandligInJava {

	public static void main(String[] args) throws IOException  {
		FileHandelling name=new FileHandelling();
		name.countWords();
		name.countLines();
		name.writeIntoAFile();
		name.countCharacter();
	}
}

class FileHandelling{
	static int count=1;
	static int lines=0;
	static int countChar=0;
	void countWords() throws IOException{
		String path="C:\\Users\\HP\\ash-eclipse-
workspace\\AshishJavaApplication\\src\\unit03\\Untitled 1";
		File file=new File(path);
		FileReader obj=new FileReader(file);
		BufferedReader br=new BufferedReader(obj);
		int prev=0,next=0;
		prev=br.read();
		while((next=br.read())!=-1)
		{
			if(prev==' '||prev=='\n'&&(next!=' '||next!='\n'))
			{
				count++;
			}
			prev=next;
		}
		System.out.println("Count Of Words:"+count);
```

```
                obj.close();
                br.close();
        }
        void countLines() throws IOException
        {
                File file=new File("C:\\Users\\HP\\ash-eclipse-
workspace\\AshishJavaApplication\\src\\unit03\\Untitled 1");
                FileReader obj=new FileReader(file);
                BufferedReader br=new BufferedReader(obj);
                String str;
                while((str=br.readLine())!=null)
                {
                        lines++;
                }
                System.out.println("Number of lines in file:"+lines);
                obj.close();
                br.close();
        }
   void countCharacter() throws IOException
   {
        File file=new File("C:\\Users\\HP\\ash-eclipse-
workspace\\AshishJavaApplication\\src\\unit03\\Untitled 1");
        FileReader obj=new FileReader(file);
        BufferedReader br=new BufferedReader(obj);
        String str;
        while((str=br.readLine())!=null)
        {
                countChar+=str.length();
        }
        System.out.println("Number of Characters:"+countChar);
        obj.close();
        br.close();
   }
}
```

Output

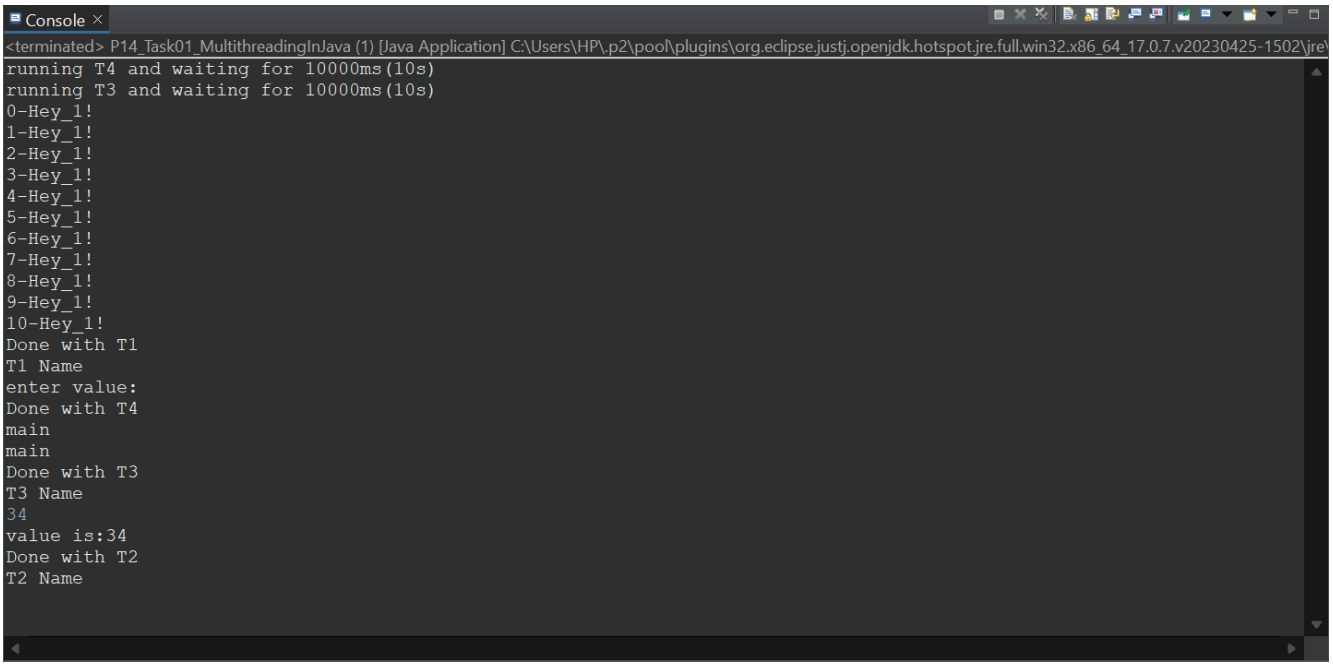

Console
<terminated> P13_Task01_FileHandligInJava [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\jav
Count Of Words:8
Number of lines in file:3
Number of Characters:37

Practical-14

```
package unit03;
import java.util.*;
public class P14_Task01_MultithreadingInJava {
	public static void main(String[] args) {
		Thread t1 = new Thread(new T1(),"T1 Name");
		Thread t2 = new Thread(new T2(),"T2 Name");
		Thread t3 = new Thread(new T3(),"T3 Name");
		Thread t4 = new Thread(new T4(),"T4 Name");
		t1.start();
		t2.start();
		t3.start();
		t4.run();
		System.out.println(Thread.currentThread().getName());
	}
}
class c1{
	void show1() {
		for(int i=0;i<=10;i++) {
			System.out.println(i+"-hey_1!");
		}
	}
}
class c2{
	void show2() {
		Scanner scan = new Scanner(System.in);
		System.out.println("enter value:");
		int a= scan.nextInt();
		System.out.println("value is:" + a);
	}
}
class T1 extends Thread{
	void display() {
		System.out.println("hello");
	}
	@Override
	public void run() {
		for(int i=0;i<=10;i++) {
			System.out.println(i+"-Hey_1!");
		}
		System.out.println("Done with T1");
		System.out.println(Thread.currentThread().getName());
```

```
	}
}
class T2 extends Thread{
	@Override
	 public void run() {
			Scanner scan = new Scanner(System.in);
			System.out.println("enter value:");
			int a= scan.nextInt();
			//scan.close();
			System.out.println("value is:" + a);
			System.out.println("Done with T2");
			System.out.println(Thread.currentThread().getName());
	 }
}
class T3 implements Runnable{
	public void run() {
		System.out.println("running T3 and waiting for 10000ms(10s)");
		try {
			Thread.sleep(10000);
		}catch(InterruptedException e) {
			e.printStackTrace();
		}
		System.out.println("Done with T3");
		System.out.println(Thread.currentThread().getName());
	}
}
class T4 implements Runnable{
	public void run() {
		System.out.println("running T4 and waiting for 10000ms(10s)");
		try {
			Thread.sleep(10000);
		}catch(InterruptedException e) {
			e.printStackTrace();
		}
		System.out.println("Done with T4");
		System.out.println(Thread.currentThread().getName());
	}
}
```

Output

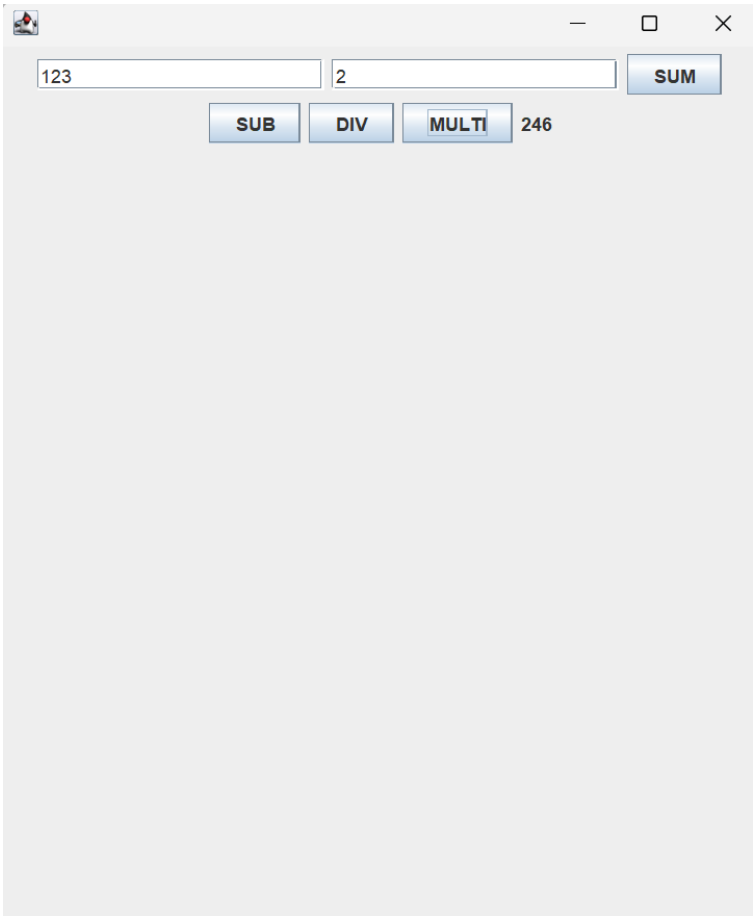

Console
<terminated> P14_Task01_MultithreadingInJava (1) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\
running T4 and waiting for 10000ms(10s)
running T3 and waiting for 10000ms(10s)
0-Hey_1!
1-Hey_1!
2-Hey_1!
3-Hey_1!
4-Hey_1!
5-Hey_1!
6-Hey_1!
7-Hey_1!
8-Hey_1!
9-Hey_1!
10-Hey_1!
Done with T1
T1 Name
enter value:
Done with T4
main
main
Done with T3
T3 Name
34
value is:34
Done with T2
T2 Name

Practical-15

```
package unit04;

import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class P15_Task01_Calculator_GUI {

	public static void main(String[] args) {
		Calculator obj=new Calculator();
	}

}
class Calculator extends JFrame{
	JTextField t1;
	JTextField t2;
	JLabel l;
	JButton b1;
	JButton b2;
	JButton b3;
	JButton b4;
	public Calculator()
	{
		t1=new JTextField(20);
		t2=new JTextField(20);
		l=new JLabel("Result");
		b1=new JButton("SUM");
		b2=new JButton("SUB");
		b3=new JButton("DIV");
		b4=new JButton("MULTI");
		add(t1);
		add(t2);
		add(b1);
		add(b2);
		add(b3);
		add(b4);
```

```
		add(l);
		ActionListener a1= new ActionListener()
		{
			public void actionPerformed(ActionEvent e)
			{
				int num1=Integer.parseInt(t1.getText());
				int num2=Integer.parseInt(t2.getText());
				//Integer result;
				if(e.getSource()==b1) {
					Integer sum=num1+num2;
					l.setText(sum.toString());
				}
				if(e.getSource()==b2) {
					Integer sub=num1-num2;
					l.setText(sub.toString());
				}
				if(e.getSource()==b3) {
					Integer div=num1/num2;
					l.setText(div.toString());
				}
				if(e.getSource()==b4) {
					Integer multi=num1*num2;
					l.setText(multi.toString());
				}

			}
		};
		b1.addActionListener(a1);
		b2.addActionListener(a1);
		b3.addActionListener(a1);
		b4.addActionListener(a1);
		setLayout(new FlowLayout());
		setVisible(true);
		setSize(500,600);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

	}
}
```

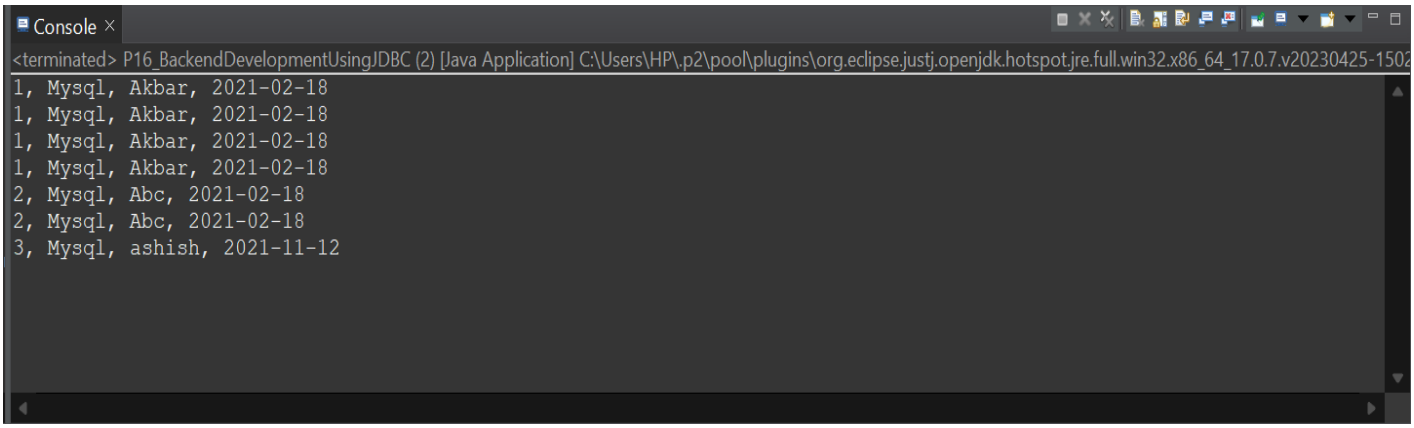
Output



Practical-16

```
package JDBC_MySql;
import java.sql.*;
public class P16_BackendDevelopmentUsingJDBC {
	public static void main(String[] args) throws SQLException {
		Connection myConn = null;
		Statement myStmt = null;
		ResultSet myRs = null;
		try {
			myConn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbcDatabase", "root" , "root");
					myStmt = myConn.createStatement();
					int a = myStmt.executeUpdate("INSERT INTO jdbc_tbl
VALUES (3, 'Mysql', 'ashish', '2021-11-12')");
						myRs = myStmt.executeQuery("select *
from jdbc_tbl");
							while (myRs.next()) {

	System.out.println(myRs.getString("jdbc_id") +
							", "
+ myRs.getString("title") + ", " + myRs.getString("author") + ", " +

	myRs.getString("submission_date"));
							}
		}
		catch (Exception exc) {
			exc.printStackTrace();
		}
		finally {
			myRs.close();
			myStmt.close();
			myConn.close();
		}
	}
}
```

Output

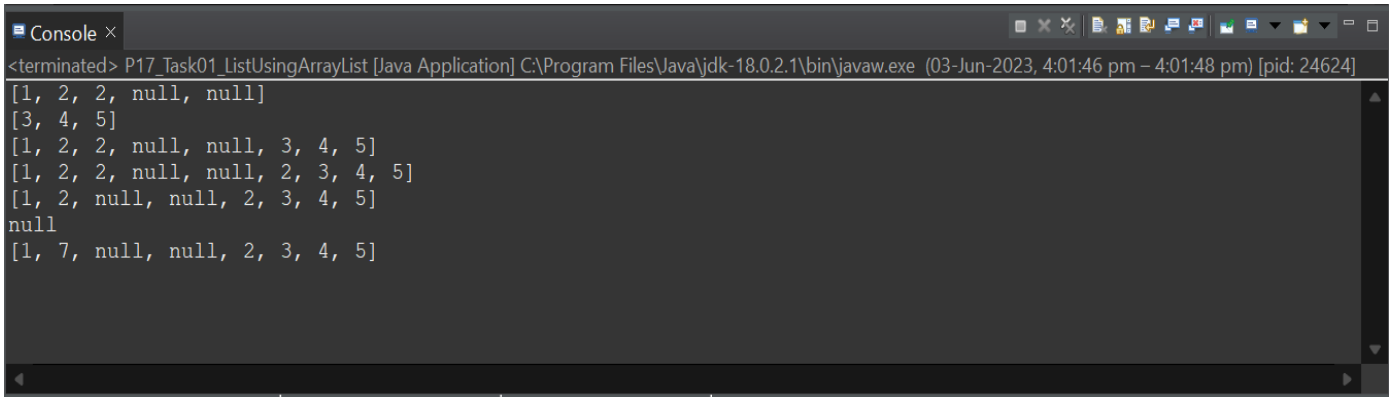

Console
<terminated> P16_BackendDevelopmentUsingJDBC (2) [Java Application] C:\Users\HP\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502
1, Mysql, Akbar, 2021-02-18
1, Mysql, Akbar, 2021-02-18
1, Mysql, Akbar, 2021-02-18
1, Mysql, Akbar, 2021-02-18
2, Mysql, Abc, 2021-02-18
2, Mysql, Abc, 2021-02-18
3, Mysql, ashish, 2021-11-12

Practical-17

```
package unit04_CollectionFrameWork;
import java.util.*;
/* List
 * don't need to give size of the array at start
 * no index out of bound exception if used properly.
 * duplicates are allowed
 * Multiple null values are allowed
 * preserve insertion order
*/

public class P17_Task01_ListUsingArrayList {

	public static void main(String[] args) {
		//creating list
		List<Integer>l1=new ArrayList<Integer>();
		l1.add(0,1);
		l1.add(1,2);
		l1.add(2,2);
		l1.add(3,null);
		l1.add(4,null);
		System.out.println(l1);
		//create another list
		//List<Integer> is an Interface while ArrayList<Integer>

		List<Integer>l2=new ArrayList<Integer>();
		l2.add(3);
		l2.add(4);
		l2.add(5);
		System.out.println(l2);

		//will add list l2 from 5 index
		l1.addAll(5,l2);
		System.out.println(l1);

		//add 2 at 5th index
		l1.add(5,2);
		System.out.println(l1);

		//remove element at index 3
		l1.remove(2);
```

```
            System.out.println(l1);

            //prints element at index 3
            System.out.println(l1.get(3));

            //replace 0th element with 5
            l1.set(1,7);
            System.out.println(l1);
        }

}
```

Output


Console
<terminated> P17_Task01_ListUsingArrayList [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (03-Jun-2023, 4:01:46 pm – 4:01:48 pm) [pid: 24624]
[1, 2, 2, null, null]
[3, 4, 5]
[1, 2, 2, null, null, 3, 4, 5]
[1, 2, 2, null, null, 2, 3, 4, 5]
[1, 2, null, null, 2, 3, 4, 5]
null
[1, 7, null, null, 2, 3, 4, 5]